CEWES MSRC/PET TR/99-19

PET Web Pages Evolution

by

Sandie Kappes

04h01899

Work funded by the DoD High Performance Computing Modernization Program CEWES Major Shared Resource Center through

Programming Environment and Training (PET)

Supported by Contract Number: DAHC94-96-C0002
Nichols Research Corporation

PET Web Pages Evolution

By Sandie Kappes, NCSA

Introduction.

The CEWES PET web site serves as a communications medium for disseminating information to the HPC user community. Enhancements to this web site were required to facilitate the assimilation, sharing and dissemination of information resources across the CEWES MSRC PET user community, and to simplify generation and maintenance of the website content.

The existing CEWES PET web site was evaluated to identify enhancements required to improve information dissemination. Recommendations were provided that incorporate state-of-the-art web technologies to facilitate information exchange within and outside of the CEWES MSRC PET user community. Selected enhancements to the PET web site were then implemented to provide an improved design utilizing state-of-the-art functionality.

This document describes the new website structure, how to migrate the old structure into the new design, and provides instructions for modifying navigation elements and utilizing Cascading Style Sheets to maintain visual consistency throughout the site.

Website structure.

The PET website uses a nested frameset architecture which consists of three main areas. These areas are shown in the image of the website shown in Figure 1 and a graphical depiction of the frameset is shown in Figure 2. The top frame, *global_nav*, contains a DoD PET image which remains consistent throughout the site. Portions of this image contain clickable regions which provide access to the MSRC Home Page, PET Home, and search functions. Since this image does not change it provides easy access to these options no matter where you are in the site. The middle frame, *subhead*, contains an image which changes according to the content displayed. There are separate images for all of the CTAs and PEIs. On the right edge of this graphic, a "Quick Find" pull down menu is provided to quickly move between CTA and PEI content featured on the site. The bottom frame, *nav_content*, consists of a navigation frame on the left, *nav*, and a content frame, *content*, on the right. This frame contains the actual information provided by the site which changes as the user peruses the site. The left navigation frame contains a hierarchical menu tree which expands to show subcategories as the user moves to the various menu items. In addition, the menu item corresponding to the content displayed is highlighted to indicate the user's location within the site.

The main advantages of this website structure are:

Visual Consistency - as a user peruses the site the general look and feel of the site remains consistent. This minimizes confusion by giving the user a sense that the information is being provided from a central source rather than from a multitude of sources. This was an important design feature for the PET website since content is provided by a number of sources.

Simplified Content Development - the overall design of the PET website is defined by the frameset architecture. It is not necessary, or recommended, that content developers spend time on website design. Rather, content developers should focus solely on development of content pages. The information included in the site can be easily developed using HTML and should primarily consist of textual information with project specific graphics. Style sheets have been provided for each area to encourage consistency in the content developed. Site navigation is provided via the navigation frame and should not generally be contained within content pages.

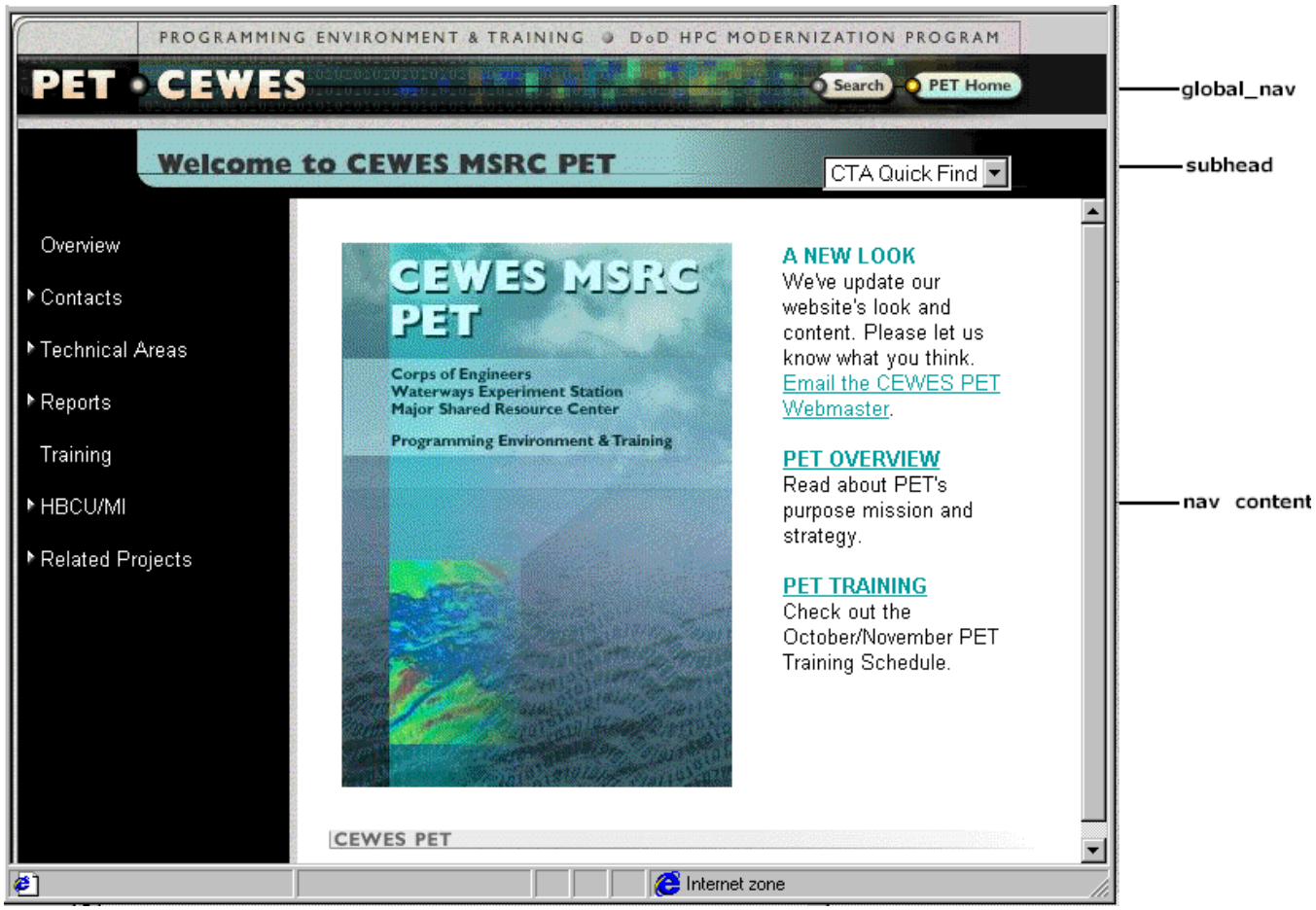

Figure 1

Visual Distinction Between Subject Areas - The CTA/PEI unique graphic in the middle frame provides visual distinction between different subject areas within the website. This enables users perusing the site to easily recognize where they are at any time.

Simple Navigation Structure - The navigation for the PET website was designed to enable the user to easily move to the desired content. All site information can be accessed from the left navigation frame, the global buttons in the top frame, or the "Quick Find" menu.

PET websites
frameset map

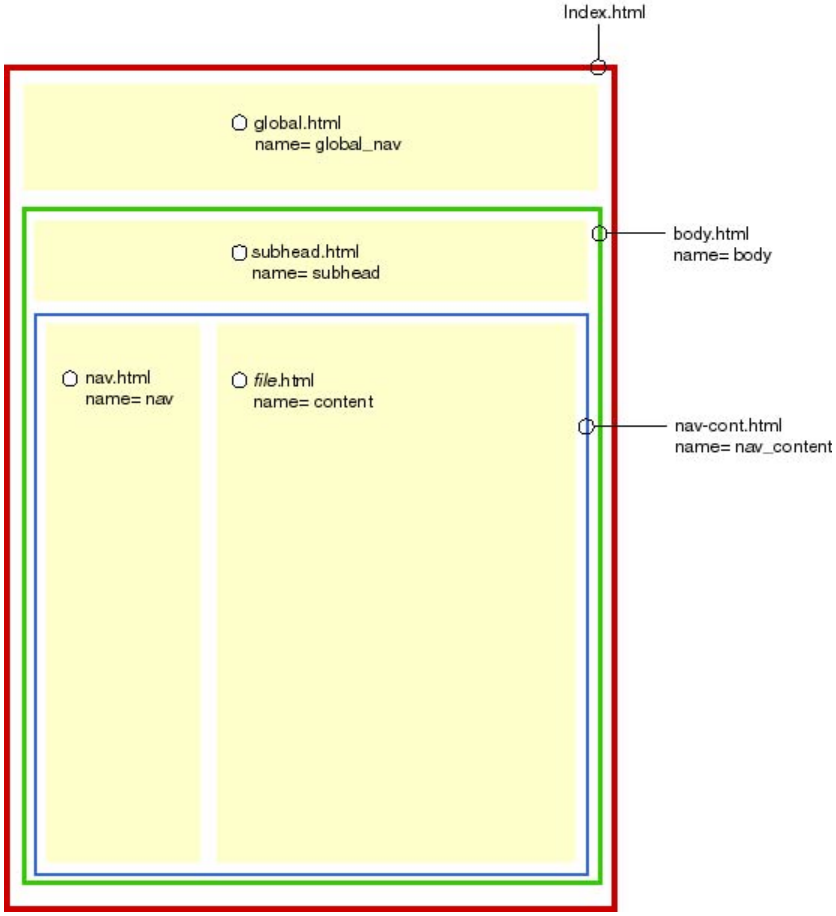

Figure 2

Migration to the New PET Web Design

The process of migrating the content from the existing PET websites to the new design is relatively straight forward. The following list describes the steps recommended for this process:

1. Reorganize Navigation Elements

There are at least 18 navigation buttons to choose from on the current ASC and CEWES sites and 25 on ARL's site. Web design guidelines suggest no more than 6 main navigation links at the top level. Reorganizing and simplifying the navigation elements and reducing redundancy whenever possible will go a long way toward making the site more navigable, legible and aesthetically pleasing.

2. Identify Additional Content Requirements

While reorganizing the navigation structure in Step 1, any additional content requirements for the site should be identified and incorporated into the navigation elements. If necessary, additional CTA/PEI information should be requested from the appropriate content providers.

3. Create Directory Structure and Populate with Files

The directory structure for the website should match the structure of the navigation elements. Once this structure is created, it should be populated with the files making up the site. The basic site provided can serve as the starting point for the directory structure.

4. Create Left Navigation Frame Contents

The left navigation frame provides an easy mechanism to change navigation within the site. Previously, modifying the navigation structure required creation of graphic buttons. This led to inconsistency across the site since some content providers developed their own navigation structure. The contents of the left navigation frame in the new design are text, not graphics, displayed via a JavaScript. The navigation elements are linked to the the page displayed in the content frame. Each navigation element can contain subcategories linked to specific elements contained in the content frame.

Navigation elements are defined in the "cats.js" JavaScript file. The contents of this file are read by the "tree.js" JavaScript file to dynamically write the contents of the nav.html file displayed in the left navigation frame. There is one "cats.js" file in the root directory and one in each of the CTA directories. The file for the associated area must be modified each time an element is added or removed (e.g. if CSM adds/removes a navigation element the "cats.js" file in the CSM directory must be modified).

A tool will eventually be provided to assist webmaster's and content providers in modifying the navigation structure. However, it is relatively easy to comprehend the syntax required in the "cats.js" file so it can be modified directly without the tool (just remember to save it as a text file!). An example "cats.js" file and instructions for defining the navigation structure is included later in this document.

5. Add Search Mechanism

A Search button has been provided in the top global frame. This button should be linked to the search engine of the site's choosing. This can be done by adding the proper elements to the "search.html" file in the **search** directory provided in the basic site.

6. Content Development and Reformatting

The additional required content as identified in step 2 should be developed in conjunction with the CTA or PEI responsible for it. Existing content in the site should be reviewed to determine if modifications are required to make it more visually pleasing and consistent with the new design. Home pages were provided in the basic site for each CTA/PEI. These are intended as a good example of a homepage structure. They contain a graphic image unique for each area and sample text for how the page could be used to direct viewers to the latest topics on the site.

Style sheets have been included in the new design. To maintain consistency across the site, it is recommended that each HTML file links to the appropriate style sheet and font tags are removed from elements targeted by the style sheet. Instructions for incorporating style sheets into the new design are given later in this document.

Instructions for adding/removing navigation elements.

The elements included in the left navigation frame are defined in a file called "cats.js". Each CTA, as well as the top level contains its own "cats.js" file. For each element, a new Category object must be created.

The best way to see how to set up the navigation structure is to look at a sample "cats.js" file. The following file creates the navigation structure:

Home
Contacts
    PET Leadership
    On-Site Staff
    Prime Contractor
    University Teams
    HBCU/MI Partners
Technology Areas
    CCM
    CEA
    CFD
    CSM
    I/C
    PT/ES
    SV
    T/C
Related Sites
    HPCMO
    MSRCS
    PET Sites

Sample "cats.js" JavaScript file:

```
1.  // -*-C++-*-

2.  root = "/ASC/";

3.  function createCats(){

4.  var cats = new Category( "menu", new tLink ("Menu") );
5.  //BEGIN CATS
6.  //
7.  cats.addChild( new tLink( "Home", "home.html", "parent.frames[1]" ) );
8.  cats.addChild( new tLink( "Contacts", "contacts/contacts.html", "parent.frames[1]" ) );
9.    cats.children[1].addChild( new tLink( "PET Leadership", "contacts/contacts.html#PET", "parent.frames[1]" ) );
10.  cats.children[1].addChild( new tLink( "On-Site Staff", "contacts/contacts.html#onsite", "parent.frames[1]" ) );
11.  cats.children[1].addChild( new tLink( "Prime Contractor", "contacts/contacts.html#nichols", "parent.frames[1]" ) );
12.  cats.children[1].addChild( new tLink( "University Teams", "contacts/contacts.html#university", "parent.frames[1]" )
     );
13.  cats.children[1].addChild( new tLink( "HBCU/MI Partners", "contacts/contacts.html#hbcu", "parent.frames[1]" ) );
14. cats.addChild( new tLink( "Technology  Areas", "techareas/techareas.html", "parent.frames[1]" ) );
15.  cats.children[2].addChild( new tLink( "CCM", "ctas/ccm/index.html", "top", true ) );
16.  cats.children[2].addChild( new tLink( "CEA", "ctas/cea/index.html", "top", true ) );
```

```
17.   cats.children[2].addChild( new tLink( "CEN", "ctas/cen/index.html", "top", true ) );
18.   cats.children[2].addChild( new tLink( "CFD", "ctas/cfd/index.html", "top", true ) );
19.   cats.children[2].addChild( new tLink( "CSM", "ctas/csm/index.html", "top", true ) );
20.   cats.children[2].addChild( new tLink( "I/C", "ctas/ic/index.html", "top", true ) );
21.   cats.children[2].addChild( new tLink( "PT/ES", "ctas/ptes/index.html", "top", true ) );
22.   cats.children[2].addChild( new tLink( "SV", "ctas/sv/index.html", "top", true ) );
23.   cats.children[2].addChild( new tLink( "T/C", "ctas/tc/index.html", "top", true ) );
24. cats.addChild( new tLink( "Related Sites", "relatedproj/projects.html", "parent.frames[1]" ) );
25.   cats.children[3].addChild( new tLink( "HPCMO", "relatedproj/projects.html#hpcmo", "parent.frames[1]" ));
26.   cats.children[3].addChild( new tLink( "MSRCS", "relatedproj/projects.html#msrc", "parent.frames[1]" ));
27.   cats.children[3].addChild( new tLink( "PET Sites", "relatedproj/projects.html#pet", "parent.frames[1]" ));

28. //END CATS

29. return cats;

30. }
```

The first Category created in line 4.,

```
var cats = new Category( "menu", new tLink ("Menu") );
```

holds all of the other categories defined. This line is always the same and should not be modified.

Top level navigation elements are created using the syntax -

```
cats.addChild( new tLink( "element title", "link", "parent.frames[1]" ) );
```

You only need to enter the value for "element title" and "link" where "element title" is the name of the navigation element and "link" is the location or name of the file to be displayed.

The navigation element "Home" which links to the file "home.html" in the root directory is defined in line 7 -

```
cats.addChild( new tLink( "Home", "home.html", "parent.frames[1]" ) );.
```

This element does not have any subcategories. Each of the top level categories are defined using the same syntax with the appropriate values entered for "element title" and "link" as shown in lines 8, 14, and 24. The order of the elements in the navigation hierarchy is determined by the order it is defined in the file.

The last three categories - Contacts, Technology Areas, and Related Sites - each have subcategories. Subcategories are defined using the following syntax:

```
cats.children["category number"].addChild( new tLink( "element title", "link", "parent.frames[1]" ) );
```

As in the top level navigation, you need to enter the value for "element title" and "link" where "element title" is the name of the navigation element and "link" is the location or name of the file to be displayed. You also need to enter a number value for "category number". This number is determined by the position in the navigation structure of it's parent element. JavaScript begins counting at 0 not 1, so for the top level categories defined in the example, the position of Home, Contacts, Technology Areas, and Related Sites is 0,1,2, and 3 respectively.

The subcategories of "Contacts" which link to a specific anchor tag in the "contacts.html" file displayed by the "Contacts" link are defined in lines 9 through 13 -

```
cats.children[1].addChild( new tLink( "PET Leadership", "contacts/contacts.html#PET", "parent.frames[1]" ) );
cats.children[1].addChild( new tLink( "On-Site Staff", "contacts/contacts.html#onsite", "parent.frames[1]" ) );
cats.children[1].addChild( new tLink( "Prime Contractor", "contacts/contacts.html#nichols", "parent.frames[1]" ) );
cats.children[1].addChild( new tLink( "University Teams", "contacts/contacts.html#university", "parent.frames[1]" ) );
cats.children[1].addChild( new tLink( "HBCU/MI Partners", "contacts/contacts.html#hbcu", "parent.frames[1]" ) );
```

Notice that the link values entered use the same format used when defining a link to an anchor tag in html. This follows from the fact that these definitions are used to write the content of the nav-cont.html file which displays the navigation menu in the left frame of the site. Hence, any valid html value can be entered for the link value in both top level and sub categories.

Note: JavaScript files are text files so don't forget to save them as such. If it is saved in the wrong format you will receive a JavaScript error during execution.

Using Cascading Style Sheets.

Cascading Style Sheets (CSS) enable a web author to specify the attributes of an HTML tag in a single page or an entire site with one command. For instance, if you want all top level headers - <h1> tag - on your site to be Helvetica 24-point and blue you can do this with one line in your style sheet:

```
H1 {  font:   24:pt Helvetica: color: blue   }.
```

This eliminates the need to specify the font and color for each occurance of the tag within your HTML files. However, font tags specified in an HTML document override definitions in a style sheet. So if you are converting your existing web pages to utilize CSS you must remove any font tags within your documents.

The following simple style sheet has been provided for the PET websites:

```
body {  font-family: Arial, Helvetica, sans-serif; text-decoration: none}
h2 {  font-family: Arial, Helvetica, sans-serif; font-size: 14pt; font-weight: bold; color: #006666; text-
decoration: none; line-height: 15pt}
h1 {  font-family: Arial, Helvetica, sans-serif; font-size: 16pt; font-style: normal; line-height: 17pt; font-
weight: bolder; color: #666666; text-decoration: none}
h3 {  font-family: Arial, Helvetica, sans-serif; font-size: 12pt; font-style: normal; font-weight: bold; color:
#009999; text-decoration: none}
```

A separate style sheet was provided for the PET web pages as well as each of the CTA/PEIs. It is recommended that these basic definitions be used throughout the site. However, individual areas can easily customize other format tags they might use by adding to their area's style sheet.

A style sheet is linked to an HTML document by inserting the following code between the the <HEAD> tag:

```
<link rel="stylesheet" href="path/filename.css">
```

where "path/filename.css" is replaced with the appropriate path and file name for your CTA or PEI style sheet.

General Text Formatting Recommendations

Flush Left Alignment - Because of the strong vertical axis created by the left-hand navigation frame, it is recommended that flush-left alignment be used on all of the PET content pages. Maintaining consistency at this

level goes a long way toward improving the legibility of the site as a whole.

Use Heading Tags to Establish Hierarchy - As mentioned above, in the *.css files included with the site, we have established font size and color values for the <H1>, <H2>, and <H3> tags. They represent a hierachical order beginning with the <H1> tag.

Use Tables for Visual Interest and Organization - Sometimes HTML tables can be employed to add visual interest and help establish hierarchy on a page. Tables should be used carefully however, avoiding in most cases the use of bright color and heavy borders. Using light, neutral color backgrounds in individual table cells is a good way to subtley distinguish columns and rows of information. See below for an example.

Project Schedule

Week 16 4/27 - 5/1	Usability study/surveys finalized Survey form pages created Write survey HTML
Week 15 5/4 - 5/8	Finalized Surveys Begin writing survey scripts
Week 14 5/11 - 5/15	Finalize scripts Send email request for participation in survey
Week 13 5/18 - 5/22	Research current site structure and develop flowchart sketch Analyze current site structure for weaknesses

Limit the Color Palette - Less is still more. Limiting the color palette on your pages will make for easier scanning and reading of the information. Using a discreet number of changes in font size, weight and color also helps bring a page into focus and makes it generally more legible.

Ubiquitous Footer Image - It is a good idea to include the PET site footer image on every content page you create, making it link back to the PET home page. This is to ensure that if a page gets separated from the site navigation via a search engine, etc. there will always be a means of getting back into the site. A footer image like the one shown below has been included for each of the PET sites:

Maintain Consistency - Whatever your design decisions, strive to maintain consistency across the entire website. Good web design is a combination of personal taste and usability guidelines so there is no one way to do it right. However, you may need to compromise your own personal tastes in order to be consistent with the rest of the site.